ASSIGNMENT-3

Support Vector Machine classifier

CS60050-Machine-Learning

Group - 1
Haasita Pinnepu (19CS30021)
Piriya Sai Swapnika (19CS30035)

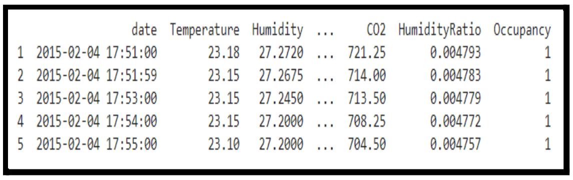
Abstract:

This report is submitted as part of the third assignment to the course CS60050 – Machine Learning, IIT Kharagpur. We used the Occupancy Detection Data Set for training the SVM task. The dataset contains experimental data used for binary classification (room occupancy) from Temperature, Humidity, Light and CO2 of 8143 instances described by 7 attributes. Our task was to use several dimensionality reduction techniques to reduce the feature dimension of data and then train Support Vector Machines. This report contains a brief overview of our approach and the results we obtained in the process.

Assessing Dataset:

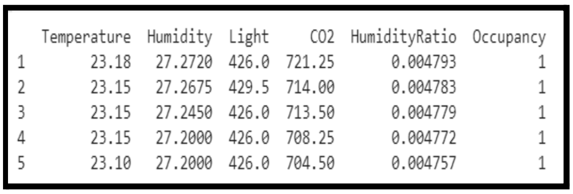
The dataset is provided to us as a text file. We accessed the data using pandas.read_csv API for the dataset handling. This is done because it is easy to handle and manipulate data using such objects. We then removed the column containing the time stamp for better assessment.

Fig1: Raw DataSet



```
                  date  Temperature  Humidity  ...     CO2  HumidityRatio  Occupancy
1  2015-02-04 17:51:00        23.18   27.2720  ...  721.25       0.004793          1
2  2015-02-04 17:51:59        23.15   27.2675  ...  714.00       0.004783          1
3  2015-02-04 17:53:00        23.15   27.2450  ...  713.50       0.004779          1
4  2015-02-04 17:54:00        23.15   27.2000  ...  708.25       0.004772          1
5  2015-02-04 17:55:00        23.10   27.2000  ...  704.50       0.004757          1
```

Fig2:DataSet after using pandas API



```
   Temperature  Humidity  Light     CO2  HumidityRatio  Occupancy
1        23.18   27.2720  426.0  721.25       0.004793          1
2        23.15   27.2675  429.5  714.00       0.004783          1
3        23.15   27.2450  426.0  713.50       0.004779          1
4        23.15   27.2000  426.0  708.25       0.004772          1
5        23.10   27.2000  426.0  704.50       0.004757          1
```

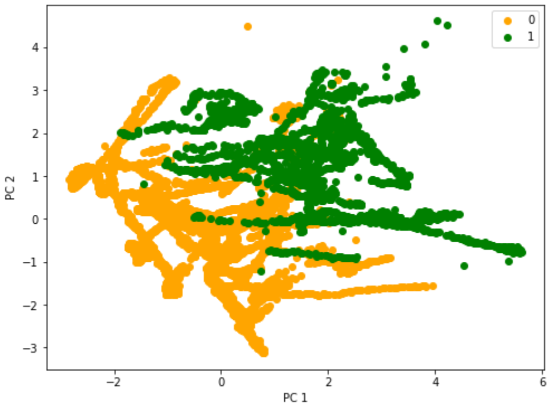
We then randomly split the dataset into train, validation and test parts. The ratio of the train, validation and test splits should be 70: 10: 20 respectively.

Applying Principal Components Analysis:

We then reduced the feature dimension of the above data into a two-dimensional feature space using Principal Component Analysis (PCA) and plotted the reduced dimensional data of the train split in a 2d plane. For PCA, we are using sklearn's PCA API.

In the plot, all data points of a single class have the same colour and data points from different classes have different colours. For the plot, we used matplotlib API.

Fig3: Scatter plot for PCA analysis



Support Vector Machine Classifier using PCA:

We trained an SVM classier (sklearn.svm.SVC) on the reduced dimensional data generated We tried different kernel types by varying the appropriate hyperparameters of the classier and computed the classification accuracy on the validation split.

```
Kernel   |  Accuracy
---------|-----------
 Linear  | 0.936284046692607
  rbf    | 0.9732490272373541
 poly    | 0.9265564202334631
```

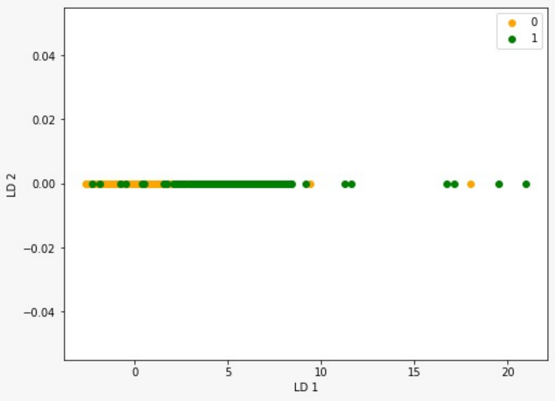
We chose rbf kernel as it has the highest accuracy.
The Test Accuracy is 0.9671772428884027

Linear Discriminant Analysis (LDA):

We reduced the feature dimension of the above data into a one-dimensional feature space using Linear Discriminant Analysis (LDA) and plotted the reduced dimensional data of the train split. For LDA, we are using sklearn's `sklearn.discriminant_analysis` API.

In the plot, all data points of a single class have the same colour and data points from different classes have a different colour.

Fig4: Scatter plot for LDA analysis



Support Vector Machine Classifier using LDA:

We trained an SVM classier (sklearn.svm.SVC) on the reduced dimensional data generated We tried different kernel types by varying the appropriate hyperparameters of the classier and computed the classification accuracy on the validation split.

```
Kernel   |  Accuracy
---------|-----------
 Linear  | 0.9888132295719845
  rbf    | 0.9892996108949417
 poly    | 0.9872461089491417
```

We chose rbf kernel as it has the highest accuracy. The Test Accuracy is `0.9878434232920009`

Result and Analysis:

Test accuracy while using PCA	Test accuracy while using LDA
0.9671772428884027	0.9878434232920009

Linear discriminant analysis is very similar to PCA both look for linear combinations of the features which best explain the data. However, we can see that the test accuracy is more for LDA when compared to PCA.

The main difference is that the Linear discriminant analysis is a supervised dimensionality reduction technique that also achieves classification of the data simultaneously. While Principal component analysis is an unsupervised Dimensionality reduction technique, it ignores the class label. LDA focuses on finding a feature subspace that maximizes the separability between the groups.PCA focuses on capturing the direction of maximum variation in the data set.